OMRON Toolbus Communication Server

for Microsoft Windows
and InTouch Applications

User Manual
Ver 1.x Rev 1.0
DR 500 10
DR 500 11

KLINKMANN AUTOMATION
P.O. Box 38
FIN-00371 Helsinki Finland
tel. int. + 358 9 5404940
fax int. + 358 9 5413541
www.klinkmann.com

Table Of Contents

Overview....1
Communication Protocols....1
Accessing Remote Items via the Server....3
Installing the OMRTOOLBUS Server....3
Installing the Server....3
Installing the I/O Server Infrastructure....5
Configuring the OMRTOOLBUS Server....5
Server Settings Command....6
Com Port Settings Command....8
Saving OMRTOOLBUS Configuration File....8
Configuration File Location....9
Topic Definition Command....9
Item Names....11
Monitoring and Controlling Communication with a PLC....15
Using the OMRTOOLBUS Server with OPC Clients....18
Configuring DCOM....18
Running OMRTOOLBUS "OPC & DDE" version as Windows NT Service....20
Using OMRTOOLBUS with OPCLink Server....21
OPCLink Topic Definition....21
Accessing OMRTOOLBUS Items via the OPCLink Server....22
Using the OMRTOOLBUS Server with Suite Link and DDE Clients....23
Using the OMRTOOLBUS Server with InTouch....23
Defining the Access Names....23
Defining the Tag names....25
Monitoring the Status of Communication with InTouch....27
Notes on Using Microsoft Excel....27
Reading Values into Excel Spreadsheets....27
Writing Values to OMRTOOLBUS Points....27
Notes on Improving Server Performance....29
Troubleshooting....31
WIN.INI entries....31
Troubleshooting menu....32
Internal Logger....34

OMRON Toolbus Communication Server

Overview

The Klinkmann Automation **OMRON Toolbus Communication Server** (hereafter referred to as the "OMRTOOLBUS Server" or "OMRTOOLBUS" or "Server") is a Microsoft Windows (95/98/ME/NT/2000/XP) 32-bit application program that acts as a communications protocol *Server* and allows other Windows application programs access to data from OMRON PLCs (also referred to as controllers or devices) using the OMRON Toolbus protocol. The OMRON Toolbus Server communicates with OMRON CS1-series and CJ1-series controllers via serial RS-232 interface, communication speed is up to 115200 bps. The *Server* may be used by any Microsoft Windows program that is capable of acting as a **DDE**, **FastDDE**, **SuiteLink** or **OPC** *Client*.

There are two different OMRTOOLBUS Server versions described in this manual:
- Server version (ordering number DR 500 10), supporting SuiteLink, FastDDE and DDE protocols; this version hereafter is referred to as the "***Suite Link & DDE***" version.
- Server version (ordering number DR 500 11), supporting OPC and DDE protocols; this version hereafter is referred to as the "***OPC & DDE***" version;

The separate installation package is supplied for each version of the Server. In all cases the name of Server executable file is **OMRTOOLBUS.EXE**. All further information in this manual is same for all versions of the Server, with the exception of few points where communication protocol specific features are explained.

Communication Protocols

Dynamic Data Exchange (DDE) is a communication protocol developed by Microsoft to allow applications in the Windows environment to send/receive data and instructions to/from each other. It implements a client-server relationship between two concurrently running applications. The server application provides the data and accepts requests from any other application interested in its data. Requesting applications are called clients. Some applications such as Wonderware InTouch and Microsoft Excel can simultaneously be both a client and a server.

FastDDE provides a means of packing many proprietary Wonderware DDE messages into a single Microsoft DDE message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between a client and a server. Although Wonderware's FastDDE has extended the usefulness of DDE, this extension is being pushed to its performance constraints in distributed environments.
The OMRTOOLBUS Communication Server supports the FastDDE Version 3 - an extension to Wonderware's proprietary FastDDE Version 2. This extension supports the transfer of Value Time Quality (VTQ) information. The original DDE and FastDDE Version 2 formats are still supported, providing full backward compatibility with older DDE clients. FastDDE Version 3 works on Windows 9x systems as well as Windows on NT/2000/XP systems.

NetDDE extends the standard Windows DDE functionality to include communication over local area networks and through serial ports. Network extensions are available to allow DDE links between applications running on different computers connected via networks or modems. For example, NetDDE supports DDE between applications running on IBM compatible computers connected via LAN or modem and DDE-aware applications running on non-PC based platforms under operating environments such as VMS and UNIX.

SuiteLink uses a TCP/IP based protocol and is designed by Wonderware specifically to meet industrial needs such as data integrity, high-throughput, and easier diagnostics. This protocol standard is only supported on Microsoft Windows NT 4.0 or higher. SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. The protocol used between a client and a server depends on your network connections and configurations. SuiteLink was designed to be the industrial data network distribution standard and provides the following features:

· Value Time Quality (VTQ) places a time stamp and quality indicator on all data values delivered to VTQ-aware clients.
· Extensive diagnostics of the data throughput, server loading, computer resource consumption, and network transport are made accessible through the Microsoft Windows NT operating system Performance Monitor. This feature is critical for the scheme and maintenance of distributed industrial networks.
· Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.
· The network transport protocol is TCP/IP using Microsoft's standard WinSock interface.

OPC (OLE for Process Control) is an open interface standard to provide data from a data source and communicate the data to any client application in a common standard way. The OPC is based on Microsoft OLE, COM and DCOM technologies and enables simple and standardized data interchange between the industrial or office sector and the production sector. From general point of view many aspects of OPC are similar to DDE, but main difference is in the implementation by using Microsoft's COM (Component Object Model) technology. It enables fast exchange with process automation data and OPC open interface allows access to data from OPC Server in same standard way from OPC client applications supplied by different developers.

For more information on the basics of OPC, please refer to the ***OPC Specification***. The OPC Data Access Custom Interface Specification is maintained by ***OPC Foundation***, the current specification is 2.05a dated June 2002 (3.00 dated March 2003).

The OPC support for OMRTOOLBUS Server "OPC & DDE" version is implemented based on ***FactorySoft OPC Server Development Toolkit*** and it conforms to OPC Data Access Custom Interface Specification 2.04. The OMRTOOLBUS Server "OPC & DDE" version is tested for compliance and is compatible with OPC Foundation OPC Data Access Compliance Test Tool.

The Suite Link, FastDDE (Version 3) and DDE support for OMRTOOLBUS Server "Suite Link & DDE" version is implemented by ***Wonderware I/O Server Toolkit*** ver. 7,2,1,6.

The FastDDE (Version 2) and DDE support for OMRTOOLBUS Server "OPC & DDE" version is implemented by ***Wonderware I/O Server Toolkit*** ver. 5.0 (008).

Accessing Remote Items via the Server

The communication protocol addresses an element of data in a conversation that uses a three-part naming convention that includes the ***application name***, ***topic name*** and ***item name***. The following briefly describes each portion of this naming convention:

application name
The name of the Windows program (server) that will be accessing the data element. In the case of data coming from or going to OMRON Controllers via this Server, the application portion of the address is **OMRTOOLBUS**.

topic name
Meaningful names are configured in the Server to identify specific devices. These names are then used as the topic name in all conversations to that device. For example, **PLC1**.
Note! You can define multiple topic names for the same device (PLC) to poll different items at different rates.

item name
A specific data element within the specified topic. When using the OMRTOOLBUS Server, an item can be a relay, timer, counter, register, etc., in the PLC. The term "point" is used interchangeably with the term "item" in this User Manual. For more information on item/point names, see the ***Item Names*** section later in this manual.

Installing the OMRTOOLBUS Server

Installing the Server

The OMRTOOLBUS Server installation package can be supplied:

1. As a self-extracting archive (50010xxx.EXE for “Suite Link & DDE” version or 50011xxx.EXE for “OPC & DDE” version) if downloaded from Klinkmann’s web site (the xxx is the current (latest) version of the Server).
2. From installation on Klinkmann Software CD.
3. On two or three distribution disks (floppies).

To **install** the OMRTOOLBUS Server from the self-extracting archive, run the 50010xxx.EXE or 50011xxx.EXE and proceed as directed by the OMRTOOLBUS Server Setup program.

To **install** the OMRTOOLBUS Server from CD or distribution disks:

1. Insert the CD with Klinkmann Software into CD drive or insert the OMRTOOLBUS Server Disk1 into a floppy drive A: or B:.
2. Select the **Run** command under the **Start** menu.
3. Run STARTUP.EXE if installing from CD or SETUP.EXE if installing from distribution disks (floppies).

4. If installing from CD: select “Protocol Servers (DDE, SuiteLink, OPC)”, find “OMRTOOLBUS SL and DDE Server” or “OMRTOOLBUS OPC and DDE Server” and click
on “Setup…”.
5. Proceed as directed by the OMRTOOLBUS Server Setup program.

Note:
All MS Windows applications using Microsoft’s shared DLLs (e.g. MFC42.DLL and MSVCRT.DLL) must be closed before installing the OMRTOOLBUS Server “OPC & DDE” version. Otherwise there can be problems with OMRTOOLBUS Server registration as OPC server. If during the OMRTOOLBUS Server “OPC & DDE” version installation some warning messages about shared DLLs are displayed, then it is quite possible the OMRTOOLBUS Server registration as OPC server failed. In this case after system reboot the OMRTOOLBUS Server registration as OPC server can be done by starting the OMRTOOLBUS Server manually with special command line parameter added: “omrtoolbus.exe /regserver”.

When installation is finished, the subdirectory specified as a folder where to install the OMRTOOLBUS Server files will contain the following files:

OMRTOOLBUS.EXE	The OMRTOOLBUS Server Program. This is a Microsoft Windows 32-bit application program.
OMRTOOLBUS.HLP	The OMRTOOLBUS Server Help file.
OMRTOOLBUS.CFG	An example configuration file.
LICENSE.TXT	Klinkmann Automation software license file.
KLSERVER.DLL	Dynamic Link Library necessary for “OPC & DDE”version of the Server.
WWDLG32.DLL	Dynamic Link Library necessary only for “OPC & DDE”version of the Server.

To **uninstall** the OMRTOOLBUS Server - start Control Panel, select “Add/Remove Programs” and select the “OMRTOOLBUS SL and DDE Server” or “OMRTOOLBUS OPC and DDE Server” from the list of available software products. Click on “Add/Remove…” and proceed as directed by the UnInstallShield program.

Notes:
1. *The OMRTOOLBUS Server “Suite Link & DDE” is developed with Wonderware I/O Server Toolkit (ver 7,2,1,6) and needs the **Wonderware FS2000 Common Components** to be installed on computer where the OMRTOOLBUS Server is running. The Wonderware FS2000 Common Components are installed automatically when any of Wonderware FS2000 Components (e.g. InTouch or some Wonderware I/O server) is installed.*
2. *If OMRTOOLBUS Server “Suite Link & DDE” version will run on PC where Wonderware FS2000 Common Components are not installed then a special **I/O Server Infrastructure installation package** can be obtained from Klinkmann Automation (see **Installing the I/O Server Infrastructure** section below). This I/O Server Infrastructure installation package contains the minimum set of software*

needed to run the OMRTOOLBUS Server "Suite Link & DDE" version and these infrastructure files must be installed prior to executing the OMRTOOLBUS Server.

3. *The HASP key is needed for full time running of OMRTOOLBUS Server. The HASP Driver setup is performed during the Server setup. Without HASP Driver installed the OMRTOOLBUS Server will run only 1 hour (with all features enabled).*

Installing the I/O Server Infrastructure

The I/O Server Infrastructure installation package can be supplied:

1. As a self-extracting archive (IOServerInfrastructure.exe) if downloaded from Klinkmann's web site or from Klinkmann Software CD.
2. On one distribution disk (floppy).

To **install** the I/O Server Infrastructure from the self-extracting archive, run the IOServerInfrastructure.exe and proceed as directed by the I/O Server Infrastructure Setup program.

To **install** the I/O Server Infrastructure from the distribution disk, on MS Windows NT:

1. Insert the I/O Server Infrastructure disk into a floppy drive A: or B:.
2. Select the **Run** command under the **Start** menu.
3. Type "A:SETUP" or "B:SETUP".
4. Click on **OK**.
5. Proceed as directed by the I/O Server Infrastructure Setup program.

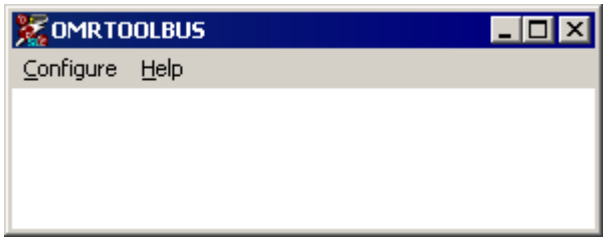
To **uninstall** the I/O Server Infrastructure - start Control Panel, select "Add/Remove Programs" and select the "IO Server Infrastructure" from the list of available software products. Click on "Add/Remove…" and proceed as directed by the UnInstallShield program.

Note: *The I/O Server Infrastructure installation will be rejected if Wonderware FS2000 Common Components are already installed on same computer.*

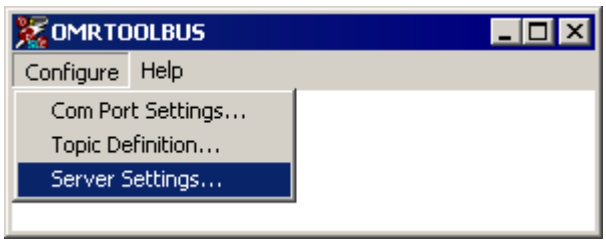
Configuring the OMRTOOLBUS Server

After the OMRTOOLBUS Server is initially installed, a little of configuration is required. Configuring the Server automatically creates an **OMRTOOLBUS.CFG** file that holds all of the topics (controllers) definitions entered, as well as the communication port configurations. This file will be placed automatically in the same directory in which **OMRTOOLBUS** is located unless the path where the configuration file will be placed is specified via the */Configure/Server Settings...* command.

To perform the required configurations, start up the OMRTOOLBUS program. If the Server starts up as an icon, double-click on the icon to open the server's window. The following will appear:



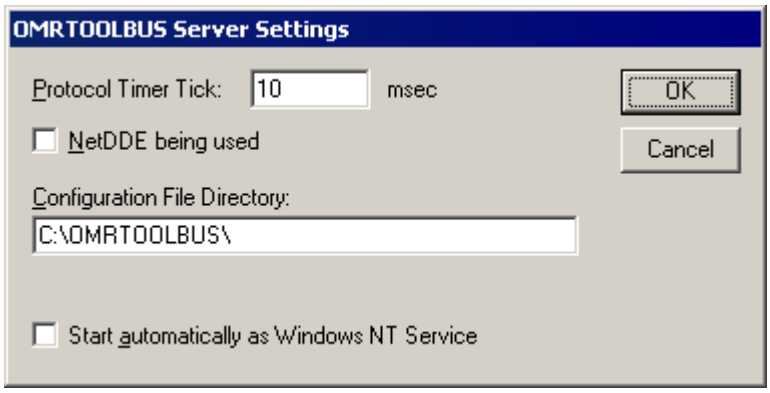
To access the commands used for the various configurations, open the */Configure* menu:



Server Settings Command

A number of parameters that control the internal operation of the Server can be set. In most cases, the default settings for these parameters provide good performance and do not require changing. However, they can be changed to fine-tune the Server for a specific environment.

To change the Server's internal parameters, invoke the *Configure/Server Settings...* command. The "Server Settings" dialog box will appear:



The following describes each field in this dialog box:

Protocol Timer Tick
This field is used to change the frequency at which the Server checks for work to do (at this frequency the Server tries to send one data request to PLC and receive one reply from PLC. If the send/response cycle is too long then more than one activation of Server is necessary to process it. If computer is very busy or some other MS Windows application is taking over the computer then the Server is activated rarely than setting in the **Protocol Timer Tick**.
Note: *The default value is 50 milliseconds. The minimum value is 10 milliseconds.*

NetDDE being used
Select this option if you are networking using NetDDE.

Configuration File Directory
The first field is used to specify the path (disk drive and directory) in which OMRTOOLBUS will save its current configuration file. OMRTOOLBUS will use this path to load the configuration file the next time it is started.

Note: *Only the "path" may be modified with this field. The configuration file is always named* ***OMRTOOLBUS.CFG.***

***Note:** There is no limit to the number of configuration files created, although each must be in a separate directory. When using the OMRTOOLBUS Server with **InTouch**, it is good practice to place the configuration file in the application directory.*

Start automatically as Windows NT Service
Enabling this option will cause the server to start as a Windows NT service.

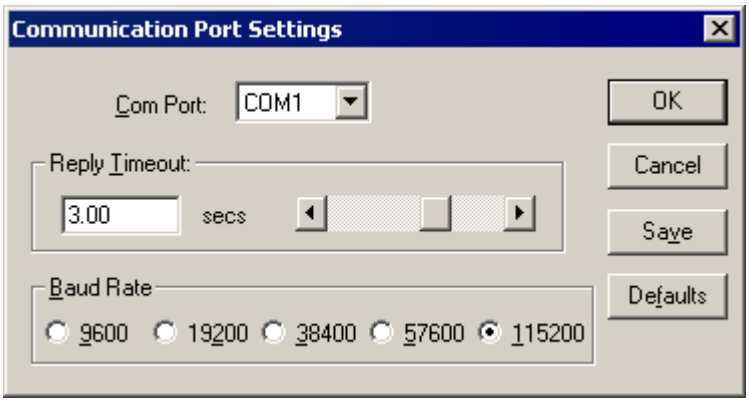
Windows NT offers the capability of running applications even when a user is not logged on to the system. This is valuable when systems must operate in an unattended mode. Enabling this option and rebooting the system will cause the Server to run as a Windows NT service. However, to view configuration information or to reconfigure the Server, the user must log on to the system. Any Server related problems that may arise such as missing adapter cards, licensing failures or device drivers not loading will not be visible to the user until a log on is performed. Disabling this option and rebooting the system will cause the Server to run as a Windows NT application program once again.

Notes:
*1. The **Start automatically as Windows NT Service** feature can be activated only with OMRTOOLBUS Server "Suite Link & DDE" version. To start the OMRTOOLBUS Server "OPC & DDE" version as Windows NT Service, refer to **Running OMRTOOLBUS "OPC & DDE" version as Windows NT Service** section of this manual.*
*2. The Service Startup configuration can be changed by MS Windows NT **Control Panel/Services** configuration dialogs. The **Allow Service to Interact with Desktop** checkbox in "Service" dialog box must be checked (the "Service" dialog box can be invoked by pressing the "Startup" button on "Services" dialog box when Service **OMRTOOLBUS_IOServer** is selected). If **Allow Service to Interact with Desktop** is not selected then OMRTOOLBUS Server full functionality is not ensured (e.g. the Server configuration can not be changed, no message boxes will be displayed, etc.).*

Once all entries have been made, click on **OK**.

Com Port Settings Command

This command is used to configure the communication ports that will be used to communicate with OMRON PLCs. Invoke the */Configure/Com Port Settings...* command. The "Communication Port Settings" dialog box will appear:



Select the **Com Port** and examine the characteristics of the selected Port. To select a necessary Com Port, click on the combo box button and make your choice from the list box.

The **Reply Timeout** field is used to enter the amount of time (in seconds) the controller using the selected communication port will be given to reply to commands from the Server. The default value of **3** seconds should be sufficient for most configurations.

Select the **Baud Rate** that matches the setting chosen for the hardware. The default Baud Rate is **115200**.

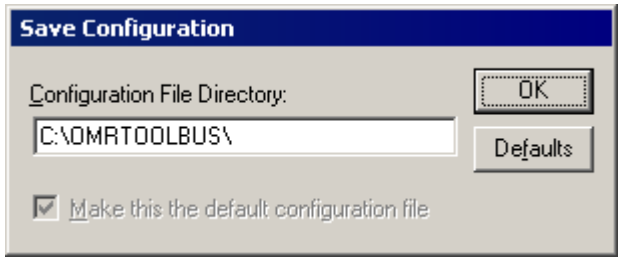
Other serial communication settings *8 Data Bits, no Parity, 1 Stop Bit* are fixed and unchangeable as specified by OMRON Toolbus protocol.

The corresponding OMRON controller must be configured for Toolbus communication and with baudrate matching with **Baud Rate** setting entered here in OMRTOOLBUS Server "Communication Port Settings" dialog – use for this purpose the OMRON CX-Programmer Windows Programming Software (see “CX-Programmer Users Manual” available from OMRON).

Once Reply Timeout and Baud Rate are entered, select **Save** to save the configuration for the communication port. Once all entries have been made, click on **OK.**

Saving OMRTOOLBUS Configuration File

If the configuration file does not currently exist, or a new configuration path has been specified, the Server will display the "Save Configuration" dialog box:



This dialog box displays the path where the Server is going to save the current configuration file. The path may be changed if necessary. Also, the path can optionally be recorded in the **WIN.INI** file by selecting the **"Make this the default configuration file"** option. Doing so will allow the OMRTOOLBUS Server to find the configuration file automatically each time it is started.

Configuration File Location

When the OMRTOOLBUS Server starts up, it first attempts to locate its configuration file by, first checking the **WIN.INI** file for a path that was previously specified. If the path is not present in the **WIN.INI** file, the Server will assume that the current working directory is to be used.

To start the Server from an application directory configuration file other than the default configuration file a special switch **(/d:)** is used. For example, invoke the **File/Run** command and enter the following:

OMRTOOLBUS /d:c:\directoryname

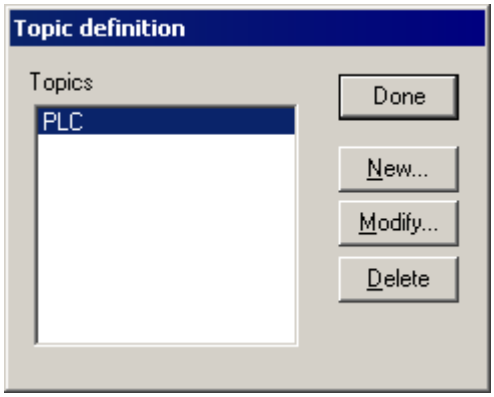
Note: *There is no limit to the number of configuration files that may be created, although each must be in a separate directory.*

Topic Definition Command

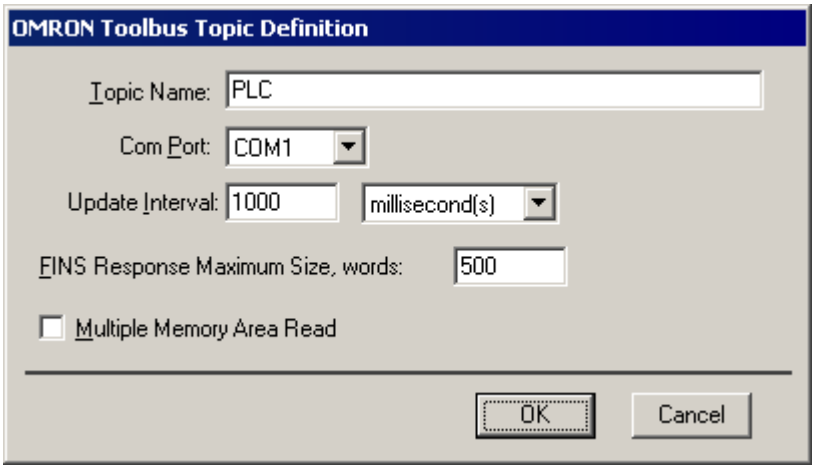
The user provides each device (controller) with an arbitrary name that is used as the Topic Name for all references to the device. Only **direct** (point-to-point) connection is supported by OMRON Toolbus protocol

The following steps are taken to define the topics (controllers) connected to OMRTOOLBUS Server:

1. Invoke the *Configure/Topic Definition…* command. The "Topic definition" dialog box will appear:



2. To modify an existing topic, select the topic name and click on **Modify**. To define a new topic, click on **New**. The "OMRON Toolbus Topic Definition" dialog box will appear:



3. Enter the **Topic Name**. If using **InTouch** the same Topic Name is to be entered in the "Add Access Name" dialog box described in the *Using the OMRTOOLBUS Server with InTouch* section.

4. Click on the **Com Port** button to associate a topic with the communication port. Additional topics may be associated with the same communication port later.

5. Set the **Update Interval** field to indicate the frequency the items/points on this topic will be read (polled). Default value is **1000** milliseconds.

6. If necessary, enter the value of **FINS Response Maximum Size, words** for *relevant data part* of read command. Values from 1 to 500 (default) are valid (if **Multiple Memory Area Read** is selected then maximum size is 250 words). The default value is recommended and reduce this value only if selected PLC has timing problems to process great amount of data in a short time period.

   All Omron CS1 and CJ1 PLCs support the requests of maximum length, but not all PLC models can process a lot of such requests without communication errors. If PLC time by time reports an error message to valid read or write command, then one of possible error reason can be too many data requested by one read command. In such cases it is recommended to reduce the **FINS Response Maximum Size, words** for the selected PLC

8. Check or uncheck the **Multiple Memory Area Read** check box to choose the type of FINS read commands used for this topic. If **Multiple Memory Area Read** is unchecked (default setting), then Server uses Memory Area Read FINS commands and maximum **500** consecutive words can be read from this topic by one command. If **Multiple Memory Area Read** is checked then Server uses Multiple Memory Area Read FINS commands and maximum **250** non-consecutive words (also from different memory areas) can be read from this topic by one command *(this mode does not support strings, long values and floating point values).*

Once all entries have been made, click on **OK.**

Item Names

The OMRTOOLBUS Server supports item/point names that are consistent with the point naming convention used by OMRON PLCs. The OMRTOOLBUS Server supports the fixed set of item names, the following describes both the valid data formats and Item Names for supported OMRON devices:

Each **Discrete**, **Integer** or **Real** (floating point) item name generally may be described as:

PnS

Each **ASCII string** (Message) item name generally may be described as:

Pn-hS

where:

n - Specifies the Memory area word/bit address of the specific word, flag or bit or lowest address of ASCII string.

The rightmost two digits of the **bit address** (when indicates a separate bit in word area) specify the bit number in word and the leftmost digits - specify the word address. Depending on the number of digits in the item name name (for bit in word leading zeroes must be in name if necessary) the Server recognizes, whether item/point represents a whole word or only a separate bit in word.

For compatibility with OMRON CX-Programmer addressing syntax and item/point naming used in some other Wonderware and Klinkmann I/O Servers, additionally also a character '**.**' (or '**:**') can be used as a separator between word address and bit address. In this case each item, representing bit in word, may be described as, **Pn.b** (or **Pn.bb** or **Pn:b** or **Pn:bb**) where **n** is a word address and **b** (**bb**) is a bit number in word (b can be 0…9 and bb can be 00…15). The number of digits in bit item name with separator is not relevant (leading zeroes are not necessary).

h - Specifies the highest address of ASCII string.

P - Item name used prefix: one, two or three characters before Memory area address, designating the type of Memory area.

S - Optionally used suffix (one character): for **Integer** and **Real** type items it specifies the special format of data (see below); for **ASCII string** items it specifies the high-order byte, low-order byte or both bytes of the words; not valid for **Discrete** type items.

Unsigned/Signed Format

The server interprets 16-bit words (channels) in a PLC in 16-bit unsigned format (value range from 0 to 65535) or signed format (value ramge from -32768 to 32767). Unsigned quantities may be read by one of two methods - by entering item name without suffix or appending the suffix '**U**' (or '**u**'). For example, the channel **DM100** in a PLC contains 8000 hex. In this case item name DM100 or DM100U would be displayed in InTouch or Excel as 32768. Signed quantities may be read from the same channel by appending the suffix '**S**' (or '**s**') to the item name. So, item name **DM100S** would be displayed in InTouch or Excel as -32768.

BCD Format

To read/write data using BCD format - append the suffix '**B**' (or '**b**') to the item name, e.g. **DM100B**.

Long Format (Signed/BCD)
Two consecutive 16-bit words in a PLC can be interpreted as a 32-bit long integer. It can be formatted as either signed (-2,147,483,648 to 2,147,483,648) or BCD (0 to 99,999,999). To read/write an item in signed long format - append the letter '**L**' (or '**l**') to the item name. To read/write an item in BCD long format - append the letter '**M**' (or '**m**') to the item name. For example, if DM100 contains 0234 hex and DM101 contains 1356 hex, **DM100L** returns 324,403,764 and **DM100M** returns 13,560,234.

Floating Point Format (IEEE/BCD)
Two consecutive 16-bit words in a PLC can be interpreted as a single precision floating point number. It is formatted as either IEEE (-3.402823 * 1038 to 3.402823 * 1038) or BCD (0.0000001 to 0.9999999 * 10^7). To read/write an item in IEEE floating point format, append one of the following letters to the item name: '**F**', '**f**', '**R**' or '**r**'. To read/write an item in BCD floating point format - append the letter '**G**' (or '**g**') to the item name. For example, if D100 contains 9620 hex and D101 contains 2436 hex in a CV500 PLC, **D100F** returns 3.959212 * 10-17 and **D100G** returns 0.4369620 * 102.
Floating point format is relevant only with items, indicating memory areas word contents.

ASCII Strings Format
Multiple consecutive 16-bit words (block size of 1 to 29) in a PLC can be interpreted as a string of ASCII characters. The ASCII string is stored/retrieved from the lowest-numbered address to the highest and from the high-order byte to the low-order byte with each address. To define a range of words, use a '-' between the two address locations. For example, DM10-DM12 indicates locations from DM10 to DM12. ASCII string characters can be specified to come from the high-order byte, low-order byte, or both bytes of the words.

Low-order Byte	Only the low-order byte (LSB - least significant 8-bits) of each word is used for read. To use this format, append the suffix '**E**' to the item name.
High-order Byte	Only the high-order byte (MSB - most significant 8-bits) of each word is used for read. To use this format, append the suffix '**D**' to the item name.
Both Bytes	Both bytes of each word are used for read/write. To use this format, append the suffix '**C**' to the item name.

Note: *All bytes in the specified memory range will be used. If the string is shorter than the range of memory specified, it will be padded with '\0'. If the string is longer than the range of memory specified, the string will be truncated.*

For example, assume the following memory contents (numeric values shown in hex):

	MSB	**LSB**	
DM10	41	42	"AB"
DM11	43	44	"CD"
DM12	45	46	"EF"

Read:

DM10 - DM12C	Returns ABCDEF
DM10 - DM11D	returns AC
DM10 - DM12E	returns BDF

Write:

DM10 - DM12C after written with "ABCD"

	MSB	**LSB**	
DM10	54	65	"AB"
DM11	73	74	"CD"
DM12	0	0	" "

DM10 - DM12E after written with "ABCDE"

	MSB	**LSB**	
DM10	0	41	"A"
DM11	0	42	"B"
DM12	0	43	"C"

Note:
Do not operate with item names, which have addresses outside really existing PLC data and memory area! *It will lead to communication failures, because the Server recognizes item name error only if item name goes out of range represented in the item names table. Usually the information like following will be logged to WWLogger and (or) to OMRTOOLBUS Internal Logger if some item with address in inaccessible memory area is used:*

"Error Response received - MRC:11 SRC:01 (first address in read command: <item name>) - wrong memory area code used or expanded memory not available"
or
"Error Response received - MRC:11 SRC:03 (first address in read command: <item name>) - the first adddress is in an inaccessible area"

Compatibility with Wonderware Omron Host Link I/O Server

The item/point naming system used in Klinkmann **OMRON Toolbus** Communication Server is compatible with item/point naming system used in **Wonderware Omron Host Link I/O Server (OMRONHL)**. In general, the item/point naming compatibility is following: the PLC types (CS1/CJ1) supported by Klinkmann OMRON Toolbus Communication Server are compatible with Wonderware Host Link Toolbus I/O Server "**CV500**", "**CV1000 and CV2000**" and "**CVM**" PLC types; the difference is in ranges of memory areas supported.

The following table lists the supported item names for the **CS1/CJ1** PLCs:

Memory Area	Prefix	Tag Type	Range	Value Range	Format Suffix
CIO Area (Bit status)	**CIO**	Discrete	**CIO000000...CIO614315** CIO0.00...CIO6143.15	0, 1	
CIO Area (Word contents)	**CIO**	Integer	**CIO0...CIO6143** 0...6143	(*)	u s b f r g l m d e c
Work Area (Bit status)	**W**	Discrete	**W000000...W051115** W0.00...W511.15	0, 1	
Work Area (Word contents)	**W**	Integer	**W0...W511**	(*)	u s b f r g l m d e c
Holding Area (Bit status)	**H**	Discrete	**H000000...H051115** H0.00...H511.15	0, 1	
Holding Area (Word contents)	**H**	Integer	**H0...H511**	(*)	u s b f r g l m d e c
Auxiliary Area (Bit status) (**)	**A**	Discrete	**A000000...A095915** A0.00... A959.15	0, 1	
Auxiliary Area (Word contents) (**)	**A**	Integer	**A0... A959**	(*)	u s b f r g l m d e c
Data Memory DM Area (Bit status)	**D** DM	Discrete	**D0000000...D3276715** DM0000000...DM3276715 D0.00...D32767.15	0, 1	
Data Memory DM Area (Word cont.)	**D** DM	Integer	**D0...D32767** DM0...DM32767	(*)	u s b f r g l m d e c
Extended Data (Bit status) (***)	**E** EM	Discrete	**E0000000...E3276715** EM0000000...EM3276715 E0.00...E32767.15	0, 1	
Extended Data (Word cont.) (***)	**E** EM	Integer	**E0...E32767** EM0...EM32767	(*)	u s b f r g l m d e c
Timer Completion Flags (****)	**T**	Discrete	**T0...T4095**	0, 1	
Counter Completion Flags (****)	**C**	Discrete	**C0...C4095**	0, 1	
Timer PVs	**PVT** PT	Integer	**PVT0...PVT4095** PT0...PT4095	(*)	u s b f r g l m d e c
Counter PVs	**PVC** PC	Integer	**PVC0...PVC4095** PC0...PC4095	(*)	u s b f r g l m d e c
Index Registers	**IR**	Long	**IR0...IR15**	(*)	m
Data Registers (*****)	**DR**	Integer	**DR0...DR15**	(*)	u s b

In **Prefix** and **Range** columns, the prefixes and range in bold are preferable.

(*) Value range for unsigned integers is from 0 to 65535 (item/point name with suffix "**U**" ("**u**") or without suffix), for signed integers from -32768 to 32767 (item/point name with suffix "**S**" ("**s**")) and for 16-bit BCD is from 0 to 9999. The value range for long or 32-bit integer is from -2147483648 to 2147483647 and for 32-bit BCD is from 0 to 99999999. Default value range for Integer Tag Type is unsigned integer.

(**) Data cannot be written to addresses A000…A447 in the Auxiliary Area.

(***) Current bank only (Memory Area code 98).

(****) Timer Completion flags and Counter Completion flags are read only.

(*****) Not supported on CJ1 controllers.

If item/point represents a separate bit in word area (relevant in **CIO**, **W**, **H**, **A, D** and **E** areas), then two conventions can be used for item/point naming:

1) Item/point names without word/bit separator are supported for compatibilty with Wonderware Hostlink I/O Server item naming. The word address in this case is specified by the leftmost five digits (**D** and **E** items) or by leftmost four digits (**CIO**, **W**, **H** or **A** items). Therefore every item name, representing the Bit, must contain 7 digits (**D** and **E** area, e.g. D0001007 is a bit 7 in D area address 10) or 6 digits (**CIO**, **W**, **H** and **A** items, e.g. H000412 is a bit 12 in H area address 4). If necessary, address in item name contains leading zeroes.
2) Character '**.**' (or '**:**') can be used as a separator between word address and bit address. For example, bit number 3 in word number 100 of CIO area can be specified as CIO100.03 (or CIO100:03).

The following examples show the correct format for item names for CS1 and CJ1 PLCs:

CIO000201	-	CIO Area (Bit), word address 2, bit number 1
CIO2.01	-	CIO area (Bit status), word address 2, bit number 1.
CIO257	-	CIO Area (Word contents), word address 257
H006210	-	Holding Area (Bit), word address 62, bit number 10
DM21	-	Data Memory Area with word address 21

Monitoring and Controlling Communication with a PLC

For each topic, there are following additional items offered by OMRTOOLBUS Server to monitor and control the communication with PLC.

STATUS

For each topic, there is a built-in discrete item that indicates the state of communication with PLC. The discrete item (**STATUS**) is set to **0** when communication fails and set to **1** when communication is successful.

Notes:

*The STATUS value will change to 0 (OFF) not immediately after first communication fault, but only after consecutive **SlowPollRetries** (see Troubleshooting/ WIN.INI entries section below in this manual) at same time when Topic changes to slow poll mode.*

From **InTouch** the state of communication may be read by defining an I/O Discrete tagname and associating it with the topic configured for the PLC and using **STATUS** as the item name.

From **Excel**, the status of the communication may be read by entering the following formula in a cell:

=OMRTOOLBUS|topic!STATUS

UPDATEINTERVAL
The **UPDATEINTERVAL** item is an Integer type Read/Write item used to access the currently set Update Interval (see Topic Definition Command). It indicates the current requested update interval (in milliseconds). The value of this item can be read through DDE, OPC or Suite Link. Client can poke new values to this item. The range of valid values is from 0 to 2147483647 milliseconds. The value of zero indicates that no items on this topic are updated. The write commands are still executed (new values written to PLC) if **UPDATEINTERVAL** value is 0.

Note*: By poking a value of zero to the* ***UPDATEINTERVAL*** *item, a client can stop all update activities on the corresponding topic without having to deactivate the items.*

MAXINTERVAL
The **MAXINTERVAL** item is an Integer type Read Only item used to access the measured maximum update interval (in milliseconds) of all items for the corresponding topic for the last completed poll cycle. The range of valid values is from 0 to 2147483647 milliseconds.

The **UPDATEINTERVAL** and **MAXINTERVAL** items can be used to tune the performance of communication.

ITEMCOUNT
The **ITEMCOUNT** item is an Integer type Read Only item used to access the number of active items in the corresponding topic. The range of valid values is from 0 to 2147483647.

ERRORCOUNT
The **ERRORCOUNT** item is an Integer type Read Only item used to access the number of active items with errors in the corresponding topic. The range of valid values is from 0 to 2147483647.

ERRORITEMS
The **ERRORITEMS** item is an Integer type Read/Write Only (unique for each topic) used to access the total number of items with invalid item names (these items are rejected by Server). The **ERRORITEMS** value can be reset by writing 0 to this item. The range of valid values is from 0 to 2147483647.

WRITECOUNT
The **WRITECOUNT** item is an Integer type Read Only item used to access the number of write commands (messages) waiting for execution. The range of valid values is from 0 to 2147483647.
For example, in following way the **WRITECOUNT** item can be used to avoid the increasing of memory occupied by not executed write commands:
- activate the hot link with **WRITECOUNT** item and start to monitor it;

- activate new write command (by poking new value) only if value of **WRITECOUNT** becomes equal to 0, e.g. all previous write commands are executed and memory occupied by them is freed.

SUSPEND

Special Read/Write Discrete Item **SUSPEND** may be used to control the communication with a separate topic. If application changes **SUSPEND** value from 0 to 1 then communication with topic is suspended. If **SUSPEND** value is changed back to 0 then communication with this topic is resumed.

Notes:

*1. If topic is suspended by setting **SUSPEND** value to 1, then Server rejects all new write values to this topic, i.e. no new write messages are created after **SUSPEND** value has changed from 0 to 1.*

*2. The **SUSPEND** item is not related with **Suspend Protocol / Resume Protocol** debugging choice available from OMRTOOLBUS Server system menu (see Troubleshooting/ Troubleshooting menu section below) – the **Suspend Protocol** suspends communication with **all** Topics, as well as in this state new write messages are still created.*

Using the OMRTOOLBUS Server with OPC Clients

The **"OPC & DDE"** version of OMRTOOLBUS Server is accessible from OPC Clients.

There are the following general steps needed to access an OPC item from OMRTOOLBUS Server:

1. Run OPC Client application and select the "OMRTOOLBUS OPC and DDE Server" from the list of available OPC Servers. If OMRTOOLBUS Server currently is not running, it will start automatically.
2. Create a new group (or topic if Wonderware OPCLink application is used).
3. If OPC Client supports the validating of items, validate the item before adding it.
4. Add the item. Depending on OPC Client it can be done in several different ways, for example:

   a) By entering separately the access path to topic name (valid topic name configured in OMRTOOLBUS Topic definition) and separately the item name.

   b) By entering the full path to item name in the format **TopicName.ItemName** where **TopicName** is the valid topic name configured in OMRTOOLBUS Topic definition.

   c) By browsing the server address space.

By default the OMRTOOLBUS Server is installed and used as a local OPC Server - both OPC Server and OPC Client reside on same computer. The OMRTOOLBUS Server can run also as a remote OPC Server – in this case OPC Server and OPC Client are located on separate computers. Accessing the remote OPC Server is same as for local OPC Server, but some DCOM (Distributed COM) configuration is required before accessing the remote OPC Server. The DCOM configuration must be done both on OPC Server and OPC Client computers.

Configuring DCOM

To access OMRTOOLBUS Server as a remote OPC Server, it is necessary to install OMRTOOLBUS Server on both (OPC Server and OPC Client) computers. (The OMRON FinsGateway software must be installed on both computers otherwise OMRTOOLBUS Server does not run.) Also the DCOM must be configured on both computers.

After Server installation the System Administrator must configure DCOM by **Dcomcnfg.exe** (or Dcomcnfg32.exe for Win9x) system tool. This utility is located in the Windows system directory – e.g. in \WinNT\system32\ or \Win9x\system\.

Below is a simple example how to configure DCOM on NT Workstations for OPC Server computer (computer name *JohnB*) and on OPC Client computer (computer name *SteveL*).

Action	**Remote OPC Server Computer name – *JohnB***	**OPC Client Computer name – *SteveL***
Install the OPC Server.	✓	✓
Run Dcomcnfg.exe	✓	✓
Invoke **Default Properties** and set **Default Authentication Level** to *(None)*, **Default Impersonation Level** to *Impersonate.*	✓	✓
Select OPC Server from **Applications** list and click on the **Properties…** button. Click on the **Location** tab, uncheck **Run application on this computer**, check **Run application on the following computer** and browse the remote computer *JohnB*.		✓
Edit **Security** settings: 1) set the following **custom access permissions**: NETWORK, SYSTEM, *SteveL*; 2) set the following **custom launch permissions**: INTERACTIVE, SYSTEM, NETWORK, *SteveL*; 3) be sure the **Default Configuration Permissions** contain SYSTEM.	✓	
Click on the **Identity** tab and select **The interactive user**.	✓	

Before starting a remote OPC Server, be sure the OPC Server computer and OPC Client computer can access each other on the network. The "Remote Procedure Call" Service should be started on OPC Server computer.

Now remote OPC Server is accessible for OPC Client on computer *SteveL*. To allow the access to more OPC Clients, configure DCOM on each OPC Client computer.

The following table contains most frequent errors when configuring DCOM.

Error message	**Possible reason**
DCOM not installed	DCOM has not been installed
Server execution failed	1) OPC Server can not be started 2) **Identity** for OPC server not properly configured 3) OPC Server is not located on a local hard disk 4) OPC Server path in registry is too long or uses LFN (Long File Names) 5) DCOMCNFG **Location** is not set to **Run on this computer**.
Class not registered	OPC Server has not been registered
RPC server is unavailable	1) **Remote Procedure Call** service is not running on the OPC Server computer 2) Invalid computer name for remote OPC Server 3) Make sure TCP/IP is installed properly
Interface not supported	1) Permission not granted to OPC Client 2) Guest account disabled
Access is denied	1) DCOM security not configured properly 2) OPC Server application not located on local hard disk 3) SYSTEM account in DCOMCNFG must have **Access**, **Launch** and **Configure** privileges
Error 80070776	Network error - TCP/IP has not been configured properly
Catastrophic failure	1) Trying to access an object before it is created 2) Unhandled exception is occurs on the OPC Server
Not enough storage	SYSTEM account in DCOMCNFG must have **Access**, **Launch** and **Configure** privileges

Running OMRTOOLBUS "OPC & DDE" version as Windows NT Service

To install OMRTOOLBUS Server "OPC & DDE" version to run as **Windows NT Service**, the OMRTOOLBUS Server must be started with command line parameter "/Service":

OMRTOOLBUS /Service

After this the "OMRTOOLBUS OPC & DDE Server" NT Service will be installed with Startup type "Manual". The Service Startup configuration can be changed by MS Windows NT *Control Panel/Services* configuration dialog boxes. The **Allow Service to Interact with Desktop** checkbox in "Service" dialog box must be checked (the "Service" dialog box can be invoked by pressing the "Startup" button on "Services" dialog box when Service **OMRTOOLBUS OPC & DDE Server** is selected). If **Allow Service to Interact with Desktop** is not selected then OMRTOOLBUS Server full functionality is not ensured (e.g. the Server configuration can not be changed, no message boxes will be displayed, etc.).

To use OMRTOOLBUS Server "OPC & DDE" version as Windows NT Service you may need to configure DCOM. For details of configuring DCOM refer to *Configuring DCOM* section of this manual. If "OMRTOOLBUS OPC & DDE Server" NT Service will be accessed only from local OPC clients (i.e. OMRTOOLBUS Server will not be used as a remote OPC Server), then **custom access** and **launch permissions** "NETWORK" are not needed.

To uninstall "OMRTOOLBUS OPC & DDE Server" NT Service, at first the Service must be stopped by *Control Panel/Services/Stop* and then OMRTOOLBUS Server must be started manually with command line parameter "/DelService":

OMRTOOLBUS /DelService

After this the OMRTOOLBUS Server "OPC & DDE" version will be still registered and accessible to OPC clients.

Using OMRTOOLBUS with OPCLink Server

The Wonderware OPCLink I/O Server (hereafter referred to as "OPCLink") is a Microsoft Windows application program that acts as a communication protocol converter and allows other Windows application programs access to data from local or remote OPC servers. OPCLink connects to OPC servers, converts client commands to OPC protocol and transfers data back to clients using DDE, FastDDE, or Suite Link protocols.

Please refer to ***Wonderware OPCLink Server and OPC Browser User's Guide*** for details how to install, start and use the OPCLink Server. The following information in this section covers only the most important points about using "OPC & DDE" version of OMRTOOLBUS Server with OPCLink Server.

OPCLink Topic Definition

The **Topic Definition** option from OPC Link Configure menu is used to create, modify, or delete OPCLink topic definitions. If OPC Link will communicate with OMRTOOLBUS Server then there must exist one or more topics defined for OMRTOOLBUS Server. There are following important fields on the "OPCLink Topic Definition" dialog box:

Topic Name
Enter a unique name (e.g. **PLC1**) for the PLC in this field. If using InTouch then same Topic Name is to be entered in the "Add Access Name" dialog box when defining the Access Names for OPCLink Server in InTouch WindowMaker.

OPC Server Name
Select the name of the OPC server (**OMRTOOLBUS.OPC_Server.1**) that will be used by this topic. The list box shows the registered OPC servers in the system.

OPC Path
Enter the name of the OPC path (e.g. **PLC1.**) used by this topic. This OPC path is the first part of a full OPC item name string common to all items that will be used in this topic. The available OPC paths for OMRTOOLBUS Server can be obtained by clicking on "Browse" button (this allows to view the OMRTOOLBUS Server's exposed address space).

Update Interval
Enter the frequency (in milliseconds) that the server will acquire data for the items/points associated with this topic. If 0 (zero) is entered here, OPCLink will not gather data from OMRTOOLBUS Server.

Browse
Clicking on this button initiates the browsing through exposed address space of OMRTOOLBUS Server. All currently available item/point names (including data area names and range of addresses) and names of pre-defined (additional) items will appear on "Browse OPC items:" window in alphabetical order.

Accessing OMRTOOLBUS Items via the OPCLink Server

The communication protocol addresses an element of data in a conversation that uses a three-part naming convention that includes the ***application name***, ***topic name*** and ***item name***. The following briefly describes each portion of this naming convention:

application name
The name of the Windows program (server) that will be accessing the data element. In the case of data coming from or going to OMRTOOLBUS Server "OPC & DDE" version, the application portion of the address is **OPCLINK**.

topic name
Meaningful names are configured to identify specific devices (PLCs). These names are then used as the topic name in all conversations to that device (PLC). This must be same name as **Topic Name** entered in the "OPCLink Topic Definition" dialog box, for example, **PLC1**.
Note! You can define multiple topic names for the same PLC to poll different points at different rates.

item name
A specific data element within the specified topic. The OPCLink Server item syntax follows the following rules. The item names must start with:

d – discrete value
i – integer value
r – real value
m – message value

The item name added to the OPC path of the topic (without the heading type letter) must give a fully qualified OPC item name for the OMRTOOLBUS Server. Some examples of possible item names acceptable by OPCLink Server/OMRTOOLBUS Server connection:

dCIO002113
discrete, CIO area (Bit status), word address 21, bit number 13
iDM100
integer, Data Memory area word 100
rDM3R
real, real floating point item, value is taken from Data Memory area words DM3 and DM4

Using the OMRTOOLBUS Server with Suite Link and DDE Clients

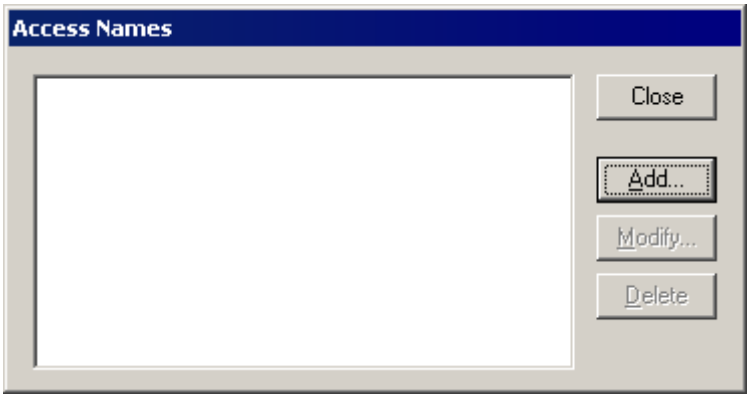
The **"Suite Link & DDE"** version of OMRTOOLBUS Server is accessible from Suite Link clients (e.g. InTouch) and DDE clients (e.g. Excel). The **"OPC & DDE"** version of OMRTOOLBUS Server is accessible from DDE clients.

Using the OMRTOOLBUS Server with InTouch

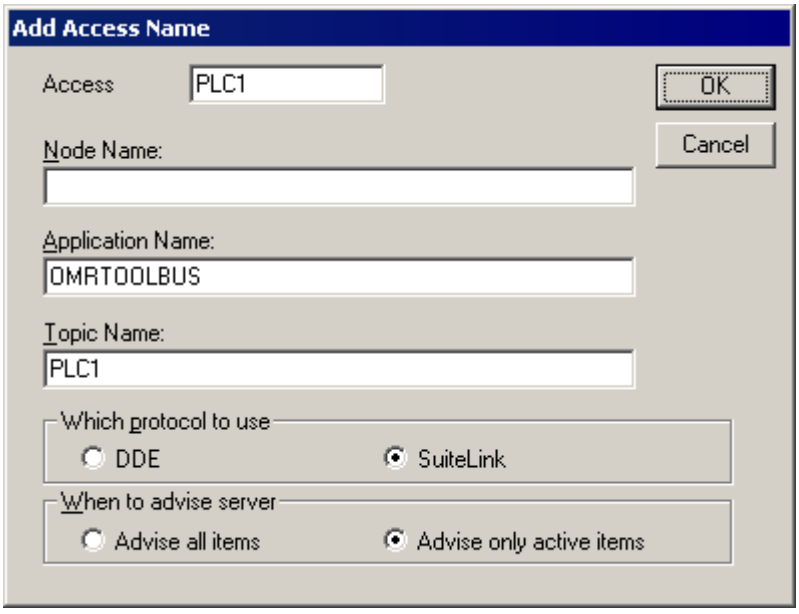
To access items/points on **Omron** PLCs from **InTouch**, the Access Names and Tag names should be defined in **WindowMaker**.

Defining the Access Names

InTouch uses **Access Names** to reference real-time I/O data. Each Access Name equates to an I/O address, which can contain a **Node**, **Application**, and **Topic**. In a distributed application, I/O references can be set up as global addresses to a network I/O Server or local addresses to a local I/O Server.
To define the Access Names in WindowMaker node invoke the */Special/Access Names...* command. The "Access Names" dialog box will appear.



Click on **Add…**. The "Add Access Name" Dialog Box will appear:



***Note:** If **Add** is selected, this dialog box will be blank when it initially appears. Data has been entered here to illustrate the entries that are made.*

The following fields are required entries when entering an Access Name Definition:
Access Name
In the Access Name box type the name you want InTouch to use to this Access Name. (For simplicity, use the same name that you will use for the **Topic Name** here.)

Node Name
If the data resides in a network I/O Server, in the Node Name box, type the remote node's name.

Application Name
In the Application Name box, type the actual program name for the I/O Server program from which the data values will be acquired. In case the values are coming from the OMRTOOLBUS Server the OMRTOOLBUS is used. Do not enter the .exe extension portion of the program name.

Topic Name
Enter the name defined for the topic in the OMRTOOLBUS Server to identify the topic the OMRTOOLBUS Server will be accessing.
The Topic Name is an application-specific sub-group of data elements. In the case of data coming from OMRTOOLBUS Server program, the topic name is the exact same name configured for the topic in the OMRTOOLBUS Server.

***Note:** This will usually be the same as the "Access Name", although, if desired, they may be different. However, it must be the same name used when the topics were configured in section **Configuring the OMRTOOLBUS Server**.*

Which protocol to use
Select the protocol (DDE or Suite Link) that you are using.

When to advise server
Select **Advise all items** if you want the Server program to poll for all data whether or not it is in visible windows, alarmed, logged, trended or used in a script. Selecting this option will impact performance, therefore its use is not recommended.

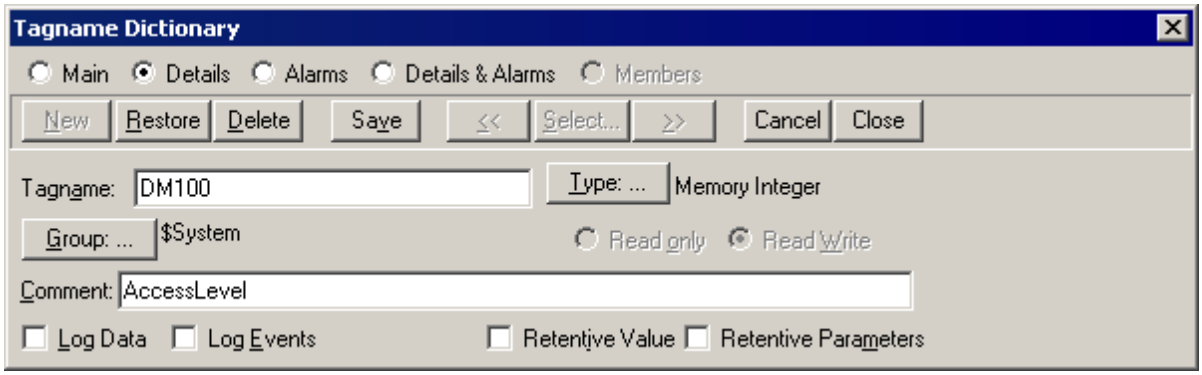
Select **Advise only active items** if you want the Server program to poll only points in visible windows and points that are alarmed, logged, trended or used in any script.

Click **OK** to accept the new Access Name and close the "Add Access Name" dialog box. The "Access Names" dialog box will reappear displaying the new Access Name selected in the list.

Click **Close** to close the "Access Names" dialog box.

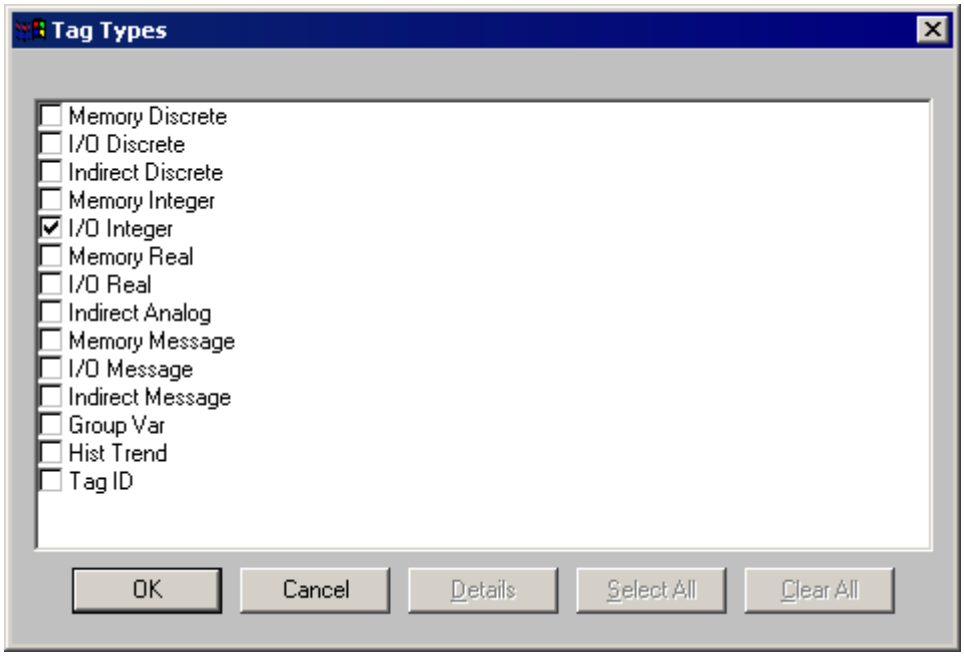
Defining the Tag names

To define the Tag names associated with the new "Access Name", invoke the */Special/Tagname Dictionary...* command (in **WindowMaker**). The "Tagname Dictionary" dialog box will appear:



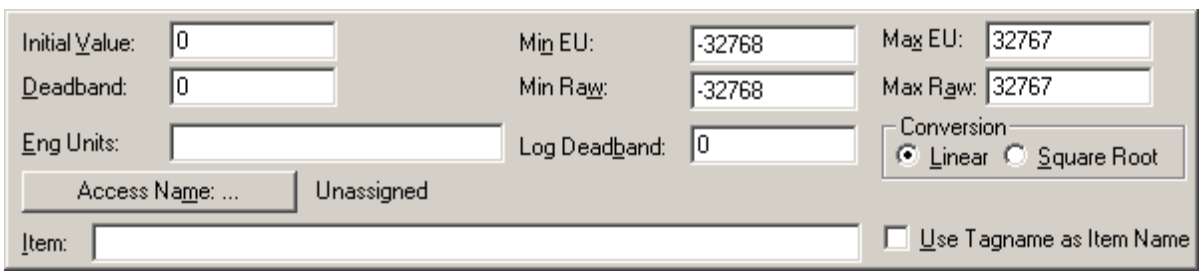
Click on **New** and enter the **Tag Name**. (The tag name defined here is the name **InTouch** will use. The OMRTOOLBUS Server does not see this name.)

Select the tag type by clicking on the **Type:…** button. The "Tag Types" dialog box will appear:

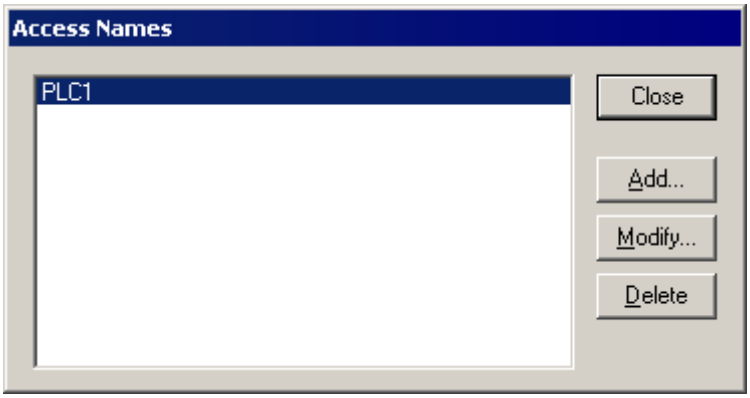


To access OMRTOOLBUS items, the type must be **I/O Discrete, I/O Integer, I/O Real** or **I/O Message.** Select the Tag type.

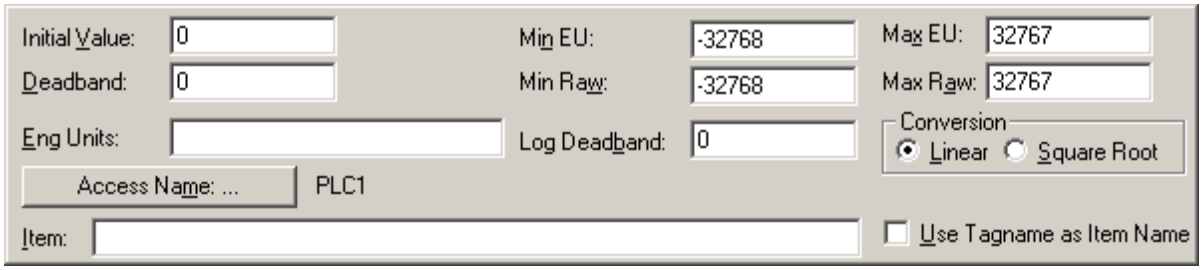
The "Details" dialog box for the tag name will appear:



Select the Access Name for OMRTOOLBUS Server by clicking on the **Access Name:…** button. The "Access Names" dialog box will appear:

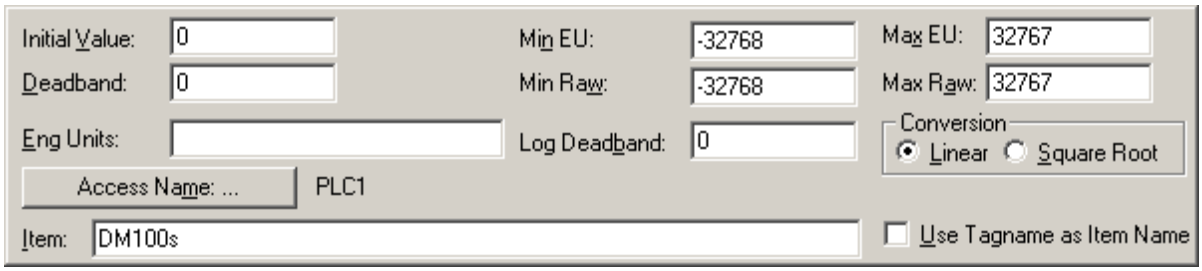


Select the appropriate Access Name and click on **Close**. (If the Access Name has not been defined as previously described, click on **Add** and define the Access Name now.) The "Details" dialog box will appear displaying the selected Access Name:



For integers and reals fill in the **Min EU**, **Max EU**, **Min Raw** and **Max Raw** fields. These fields control the range of values, which will be accepted from the Server and how the values are scaled. If no scaling is desired, **Min EU** should be equal to **Min Raw** and **Max EU** equal to **Max Raw**.

Enter the OMRTOOLBUS item name to be associated with this tagname in the **Item:** field in the "Details" box:



(Refer to the **Item Names** section below for complete details.)

Where applicable, the **Use Tagname as Item Name** option may be selected to automatically enter the tag name in this field. ***Note:*** *The tag name can only be used if it follows the conventions listed in the* ***Item Names*** *section.*

Once all entries have been made, click on the **Save** button (in the top dialog box) to accept the new tagname. To define additional tagnames click on the **New** button. To return to the **WindowMaker** main screen, select **Close**.

Monitoring the Status of Communication with InTouch

InTouch supports built-in topic names called **DDEStatus** and **IOStatus**, which are used to monitor the status of communication between the Server and InTouch. For more information on the built-in topic names DDEStatus and IOStatus, see your online "InTouch User's Guide".

The status of communication between the Server and InTouch can be read into **Excel** by entering the following DDE reference formula in a cell on a spreadsheet (in following examples **PLC1** is the Topic Name configured for OMRTOOLBUS Server):

=view|DDEStatus! PLC1
or
=view|IOStatus! PLC1

Notes on Using Microsoft Excel

Data from OMRTOOLBUS topics (nodes) may be accessed from Excel spreadsheets. To do so, enter a formula like the following into a cell on the spreadsheet.

=OMRTOOLBUS|topic!item

Sometimes, Excel requires the topic and/or item to be surrounded by apostrophes.

In the formula, **topic** must be replaced with one of the valid topic names defined during the Server configuration process. Replace **item** with one of the valid item names described in the **Item Names** section.

Reading Values into Excel Spreadsheets

Values may be read directly into Excel spreadsheets by entering a DDE formatted formula into a cell, as shown in the following examples:

=OMRTOOLBUS|'PLC1'!'cio005515'
=OMRTOOLBUS|'CS1-3'!'SA1000'
=OMRTOOLBUS|'node10'!'IR2'

Note: *Refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.*

Writing Values to OMRTOOLBUS Points

Values may be written to the Server from Microsoft Excel by creating an Excel macro that uses the **POKE** command. The proper command is entered in Excel as follows:

channel=INITIATE("OMRTOOLBUS","topicname")
=POKE(channel,"itemname", Data_Reference)
=TERMINATE (channel)
=RETURN()

The following describes each of the above **POKE** macro statements:

channel=INITIATE("OMRTOOLBUS ","topicname")
Opens a channel to a specific topic name (defined in the Server) in an application with name OMRTOOLBUS (the executable name less the .EXE) and assigns the number of that opened channel to **channel.**

Note: *By using the* ***channel=INITIATE*** *statement the word* ***channel*** *must be used in the* ***=POKE*** *statement instead of the actual cell reference. The* ***"applicationname"*** *and* ***"topicname"*** *portions of the formula must be enclosed in quotation marks.*

=POKE(channel,"itemname", Data_Reference)
POKEs the value contained in the **Data_Reference** to the specified item name (actual location in the Omron PLC) via the **channel** number returned by the previously executed **INITIATE** function. **Data_Reference** is the row/column ID of the cell containing the data value. For **"itemname"**, use some of the valid item names specified like described in the ***Item Names*** section.

=TERMINATE(channel)
Closes the channel at the end of the macro. Some applications have a limited number of channels. Therefore they should be closed when finished. **Channel** is the channel number returned by the previously executed **INITIATE** function.

=RETURN()
Marks the end of the macro.

The following is an example of Excel macro used to poke value from cell B2 to topic **PLC1** item **DM100**:

```
PokeMacro -Ctrl a
=INITIATE("OMRTOOLBUS","PLC1")
=POKE(A2,"DM100",B2)
=ON.TIME(NOW()+0.01,"TerminateDDEChannel")
=RETURN()

TerminateDDEChannel
=TERMINATE(A2)
=RETURN()
```

Note: *Refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.*

Notes on Improving Server Performance

1. Consecutive addressing of OMRTOOLBUS items
To increase the performance of OMRTOOLBUS Server, it is highly recommended to use the consecutive addressing of items. For example, access to two consecutive channels (words) in the PLC only takes about 15% longer than access to only one channel. If addresses of channels are widely separated, the access to two channels may take up to twice long as access to only one channel.

For access to discrete items within channels, the OMRTOOLBUS Server must access to whole channel (word) to read one bit. Therefore, access up to 16 bits will take no longer as access to one bit if all 16 bits are in the same channel (word).

2. Writing bits
There is no Toolbus command available for **direct bit writing** to Omron PLCs. To write a bit value into CIO, W, H, A, D and E areas for CS1 and CJ1 PLCs, the Server at first reads a word (including the corresponding bit) from the PLC. Then according to the new bit value to be written this received word is modified and written back to the PLC. It means that 16 bits are written to PLC instead of 1 bit. If PLC program modifies this word contents between above mentioned read and write commands, it can cause incorrect execution of PLC program. It is recommended to write the whole word (where it is possible) instead of separate bit writing - it will improve the performance of the Server.

Note: *The results will be unpredictable if you use the OMRTOOLBUS Server to write to discretes which are also being used in a ladder logic program. For better results, avoid this practice.*

3. The using of PLC memory areas
The OMRTOOLBUS Server uses FINS commands to exchange information with OMRON controllers and depending on client application there are two possible choices for each topic (controller):

(1) If **Multiple Memory Area Read** (see *Topic Definition Command*) is not used then maximum 500 consecutive words can be read from controller by one read command. The Server uses the following system to create read commands: the words 0...499 are in the memory area's first message, the words 500...999 in the second, etc. The data only from one memory area can be read by one read command. In this case the optimal using of PLC memory is very important - there must be as few as possible commands sent to the controller and it is strongly recommended to use consecutive memory addresses for items/points - this will considerably improve the performance of the Server. This choice is recommended if large amount of consecutive data must be read from few memory areas.
To find out how many messages are actually sent to network the commands **Dump Screen** or **Dump** from System Menu can be used (Refer to the **Troubleshooting menu** section).

(2) If **Multiple Memory Area Read** is used then maximum 250 non-consecutive words (also from different memory areas) can be read by one command (this mode does not

support strings, long values and floating point values). This choice is recommended if non-consecutive data must be read from different memory areas.

4.Some suggestions for processing of alarms (Discrete values)
The following actions are recommended if different time intervals must be used to read Integer and Discrete values:
- create two different topics for the same PLC and use greater "Update Interval" for topic with Integers (for example 3000 milliseconds) and smaller "Update Interval" for topic with Discretes (for example 500 milliseconds);
- if possible then put together all Discretes in one memory area at consecutive addresses (may be in the PLC program) to minimize number of read messages sent at smaller "Update Interval".

5.Other suggestions
For great InTouch applications it is recommended to decrease the amount of log information used for historical trending - historical trends are stored on the disk and therefore disk read/write operations will take more time as permissible. If there are problems with the speed of screen redrawing then possible solutions would be the decreasing of total amount of redrawn information (specially texts) or the increasing of Server timing values "Protocol Timer Tick" and "Update Interval".

Troubleshooting

WIN.INI entries

The first time you run the OMRTOOLBUS Server configuration, most of the items in the following list will automatically appear in the WIN.INI file, located in the MS Windows system directory (e.g. C:\WINNT). It is an ASCII file and can be altered manually if you wish with any text editor, e.g., MS Windows Notepad (*do not use a program that formats text, such as MS Word or Write unless the file is saved as a DOS text*). The following is a typical entry for the OMRTOOLBUS Server:

[OMRTOOLBUS]
ConfigurationFile=C:\OMRTOOLBUS
WinIconic=0
WinFullScreen=0
WinTop=112
WinLeft=0
WinWidth=200
WinHeight=168

There are following additional WIN.INI entries available for the OMRTOOLBUS Server:

SlowPollRetries and SlowPollInterval
The **SlowPollRetries** entry is used to enter the number of consecutive error retries for one topic (controller). If after **SlowPollRetries** there is still no successful response from PLC, then this topic is changed to *slow poll mode*. The WIN.INI file **SlowPollInterval** entry is used to enter the slow poll mode update interval (in seconds).

The default values (they are used if WIN.INI file does not contain these entries) are **SlowPollRetries** equal to 3 and **SlowPollInterval** equal to 30 seconds.

For example, the following entries can be used to specify that slow poll mode 2 minutes will start after 5 consecutive unsuccessful retries:

SlowPollRetries =5
SlowPollInterval=120

Entering into slow poll mode is reported to WWLogger and (or) to OMRTOOLBUS Internal Logger by following string like following:

"Set slow poll mode - poll after each 120000 msecs. Stop error logging to Topic 'PLC1' (COM1)."

Leaving the slow poll mode is reported to WWLogger and (or) to OMRTOOLBUS Internal Logger by following string:

" Leaving slow poll mode on topic "PLC1", port "COM1"."

Note!
*Tthe built-in discrete item **STATUS** value will change to 0 (OFF) simultaneously when Topic is changed to slow poll mode.*

WriteRetryIndefinitely and ConsecutiveWriteErrors
The **WriteRetryIndefinitely** entry is used to enable or disable the deleting of pending write messages if PLC does not respond or respond with error response (for example, if trying to write into inaccesoble data or memory area in the PLC). As default (no **WriteRetryIndefinitely** entry in WIN.INI file or **WriteRetryIndefinitely=0**) all write messages for this topic will be deleted after **ConsecutiveWriteErrors** retries. The default **ConsecutiveWriteErrors** value is **3** (no **ConsecutiveWriteErrors** entry in WIN.INI file or **ConsecutiveWriteErrors =3**). If the total amount of write messages in the list of pending write messages exceeds 300 then every write message is sent to PLC only once not taking into account the **ConsecutiveWriteErrors** value.

To disable the deleting of pending write messages, the following entry can be used:

WriteRetryIndefinitely=1

In this case all pending write messages will wait for execution forever.

***Note!** Be careful when using the **WriteRetryIndefinitely=1** setting if PLC is switched off, but client application continues to generate new values to be written to this PLC - it can cause the computer memory oveload with memory allocated for write messages.*

Troubleshooting menu

The following debugging choices are appended to the Server's System Menu (the menu that appears when you click on the "-" box in the upper left hand corner of the Server window):

Suspend Protocol / Resume Protocol - these choices permit you to turn protocol processing on and off, what means that you can suspend access to the PLCs; in **Suspend Protocol** state communication with **all** Topics is suspended (no read or write commands are sent to connected devices), but new write messages are still created for later execution when communication will be resumed.

Show Send - if checked then all outgoing data is displayed and logged.

Show Receive - if checked then all incoming user data is displayed and logged.

Show Errors - if checked then all information about errors is displayed.

ShowEvents - if checked then Server logs the information about situations when the total number of active (pending) write messages exceeds 300. It may happen when the Server has timing problems when executing write commands. When such warnings appear then it is recommended to modify the Server timing settings or to reduce the total amount of write data in the client application. Otherwise the possibility of loosing write data increases. Also information about suspended and resumed topics is displayed if **ShowEvents** is checked.

ShowBadWrites - if checked then Server displays information about all situations when data write commands are not executed successfully and after **ConsecutiveWriteErrors** retries the write is rejected and write

message is deleted from list of active (pending) write messages. This option (if checked) is effective even if other error logging is stopped.

ShowAllRejectedWrites - this option is useful when communication with a separate topic (topics) is suspended by using SUSPEND item (see *Item Names* section) and the Server rejects each write to this topic (topics). If checked then information about each rejected write value is reported to WWLogger and (or) to OMRTOOLBUS Internal Logger. If not checked then Server rejects each write to suspended topic(s) without logging any information.

Verbose - if checked then all information about errors (except errors indicated in other choices) is displayed. This option is useful for getting additional information about error situations:

When this option is checked simultaneously with the option **Show Errors** then outgoing user data as well as incoming user data is displayed in addition to the error report. For some types of errors the Server gives explanation of possible error reason.
If client application closes communication with a topic then the Server immediately deletes all this topic's messages from message lists. If write messages are still in the list of active (pending) messages these write messages are deleted without trying to execute them. If **Verbose** option is checked simultaneously with the **ShowBadWrites** option then information about not performed and deleted write messages is displayed.

Show Logger - if checked then OMRTOOLBUS Internal Logger is activated/deactivated (see *Internal Logger* section below).

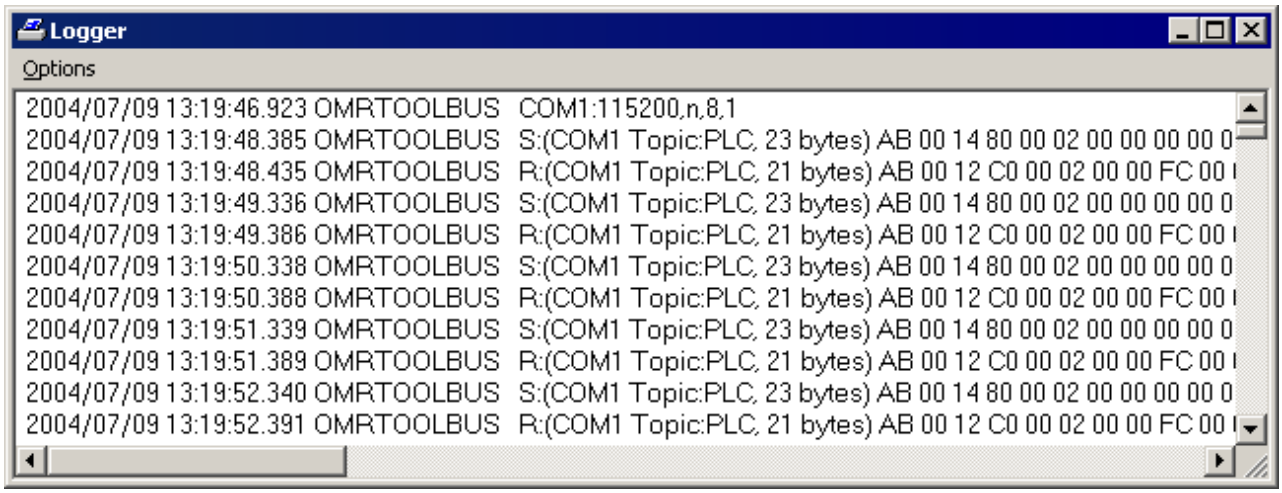
Dump - displays and logs all information about ports, active topics and items.

DumpScreen - if checked information about active topics and messages is displayed on OMRTOOLBUS main window.

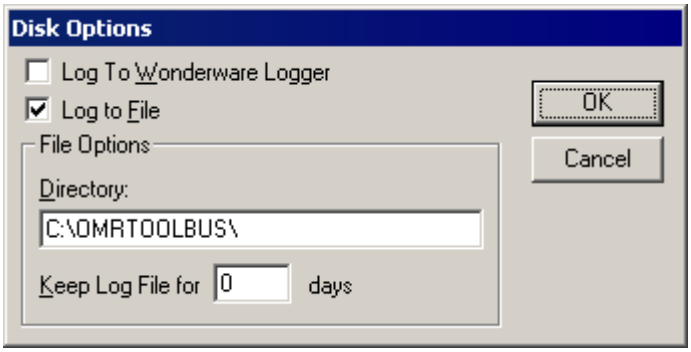
All debugging information (except **DumpScreen** option) is displayed via the Wonderware Logger (and also to OMRTOOLBUS Internal Logger if **Show Logger** checked), which must be active for these commands to work.
Warning: if you check **Show Send** and/or **Show Receive** debug output grows very fast.

Internal Logger

To enable the OMRTOOLBUS Internal Logger, check the **Show Logger** option at the OMRTOOLBUS Server System Menu (see *Troubleshooting menu* section above) - this command can be used to start/stop Internal Logger. The Internal Logger window looks like following:



To save Internal Logger information to file, select *Options/Disk Options…* from Internal Logger main menu – the “Disk Options” dialog box will appear:



The following can be entered in this dialog box:

Log to Wonderware Logger
If checked then debug information will be logged also to Wonderware Logger (if Wonderware Logger is installed).

Log to File
If checked then Internal Logger information will be saved to Internal Logger File. The OMRTOOLBUS Internal Logger file name is created in the following format:

OMRTOOLBUS _YYYYMMDD.LOGn

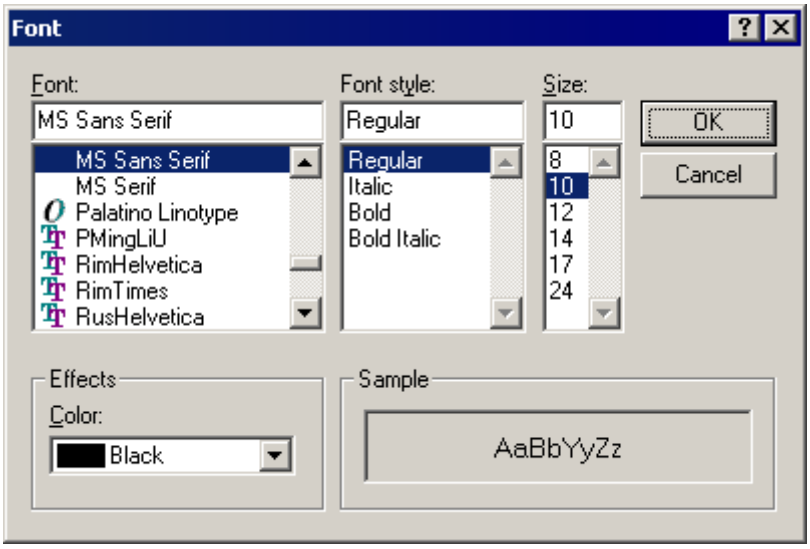
where **YYYY** is a year, **MM** is a month, **DD** is a day and **n** is a order number of consecutive OMRTOOLBUS Internal Logger file, starting from 1. The OMRTOOLBUS Internal Logger file maximum size is 16 MB; if there is more information logged then next

consecutive file is created, e.g. there can be consecutive files OMRTOOLBUS_20030228.LOG1, OMRTOOLBUS_20030228.LOG2, etc.

Directory
Enter the path where to keep the Internal Logger File.

Keep Log File for
Here the number of days how long to keep the Internal Logger File can be entered. After this number of days expires, the corresponding Internal Logger File will be automatically deleted. The default value **0** keeps Internal Logger Files forever - in this case they can be deleted manually.

Options/Font
To configure the font used by Internal Logger, select *Options/Font…* from Internal Logger main menu - the "Font" dialog box will appear:



KLINKMANN AUTOMATION
OMRON Toolbus Communication Server
Revision History

Jul 2004	Rev 1.0	First Release